8Г

УДК 681.322

С.А. Горенко, А.Л. Головинский, С.Г. Рябчун
Институт кибернетики им. В.М. Глушкова НАН Украины, г. Киев, Украина
icybcluster@gmail.com

Построение суперкомпьютера кластерной архитектуры без использования сети Ethernet

Исследованы общие концепции управления высокопроизводительной кластерной вычислительной системой без использования сети Ethernet, а также рассмотрена возможность практической реализации такого управления.

Введение

Производительность современных вычислительных узлов вполне соизмерима с производительностью небольших кластеров недавнего времени. Такой уровень вычислительных мощностей, а также потребность в управлении большим количеством вычислительных узлов в одном отдельно взятом кластере на порядок увеличивает требования к средствам обмена данными. Следовательно, использование только сети *Ethernet* не может обеспечить нужной скорости обмена данными между приложениями по протоколу *MPI* [1] для полномасштабного учета характеристик аппаратуры. Поэтому в суперкомпьютерах стало нормой использование высокоскоростных сетей (*InfiniBand*, *MyriNet*, *SCI* и др.) [2]. Они обеспечивают производительность, необходимую для полной реализации потенциала как программных, так и аппаратных средств, и могли бы заменить *Ethernet* не только как средство для обмена *MPI* сообщениями, но и для передачи любых данных. Однако, учитывая все аспекты эксплуатации кластерных вычислительных систем, управление ими без использования *Ethernet* было практически невозможно и в большинстве случаев приходилось использовать сразу две сети одновременно, хотя избыточное количество сетевого оборудования негативно влияет на такие характеристики системы, как стоимость, сложность и надежность. В этой ситуации управляющие системные функции распределяются между двумя сетями, во многом они дублируют друг друга, но в то же время ни одна из них не может в полной мере заменить вторую. Вполне логичным было возникновение идеи построения управления вычислительной системой без использования сети *Ethernet*.

Функции сети в кластерной вычислительной системе

Если мы хотим оставить только одну сеть, то необходимо обеспечить реализацию всей функциональности управления работой современной высокопроизводительной кластерной вычислительной системы. Упрощенно группами таких функций являются:

1. **Обмен данными по интерфейсу MPI**. Message Passing Interface (интерфейс передачи сообщений) – программный интерфейс (API) для передачи информации, позволяющий осуществлять обмен сообщениями между компьютерами, выполняющими параллельную задачу. Он является наиболее распространенным стандартом интерфейса обмена данными в параллельном программировании. Программное обеспечение современных кластеров чаще всего выполняет обмен данными именно через MPI.

2. **Пересылка IP пакетов**. Сетевые сервисы, необходимые для работы кластера, используют этот протокол для передачи информации по сети. Например, корневая файловая система узлов обычно монтируется с помощью Network file system (NFS), которая работает в соответствии со стеком протоколов TCP/IP, а следовательно, поддержка TCP/IP является необходимым условием функционирования кластерного комплекса.

3. **Удаленная загрузка узлов**. Учитывая большое количество вычислительных узлов в высокопроизводительных кластерах, устанавливать на каждый из них отдельную операционную систему представляется достаточно сложным, а в случае применения бездисковых узлов и невозможным процессом. Чаще всего в кластерных комплексах реализуется удаленная загрузка одного экземпляра операционной системы. Такой подход значительно упрощает администрирование вычислительной системы.

4. **Удаленное управление аппаратными средствами кластера, не зависящее от операционной системы**. Как известно, кластерные вычислительные системы чаще всего находятся на значительном расстоянии от пользователей и даже администраторов, поэтому любые простые операции, требующие непосредственного доступа к оборудованию, например, перезагрузка «зависшего» узла, без наличия специальных средств, превращается в проблему. Для упрощения обслуживания таких систем используют специальные средства, позволяющие многие подобные операции выполнять удаленно.

Полноценная работа современного высокопроизводительного кластера требует реализации как минимум упомянутых 4 групп функций и, если ставится цель отказаться при этом от использования *Ethernet*, то оборудованием всего одной сети. Среди существующих сейчас высокоскоростных сетей наиболее интересной, в контексте данной темы, является *InfiniBand* [3]. Причины такого интереса:

1. **Высокая пропускная способность**. Для передачи данных в *InfiniBand* применяются 4-проводные двунаправленные соединения. Базовая пропускная способность составляет 2,5 Гбит/с в каждом направлении при использовании Single Data Rate (SDR), поддерживается также работа с Double Data Rate (DDR) – 5 Гбит/с и Quad Data Rate (QDR) – 10 Гбит/с. Сетевые платы и коммутаторы имеют порты 4х, скорость при этом составляет соответственно 10, 20 или 40 Гбит/с.

2. **Многоплановая функциональность**. *InfiniBand* поддерживает множество протоколов, среди которых:

– *Remote Direct Memory Access (RDMA)* [3] – группа протоколов удалённого прямого доступа к памяти, при котором передача данных из памяти одного компьютера в память другого компьютера происходит без участия операционной системы и использования ресурсов центрального процессора;

– *SCSI RDMA Protocol (SRP)* [3] – протокол обмена данными между *SCSI* устройствами с использованием *RDMA*;

– *IP over InfiniBand (IPoIB)* [3] – группа протоколов, описывающих передачу IP-пакетов через *InfiniBand*;

– *Socket Direct Protocol (SDP)* [3] – протокол установления виртуальных соединений и обмена данными между сокетами через *InfiniBand*.

Кроме поддержки многих протоколов имеются также многие программные средства, позволяющие расширить возможности применения данной сети. *Boot over InfiniBand (BoIB)* [3] является одним из них и позволяет осуществлять удаленную загрузку операционной системы в сетях *InfiniBand*, что до недавнего времени было возможным только при наличии *Ethernet*.

3. **Интенсивность развития**. *InfiniBand* является довольно востребованной и активно развивающейся технологией. Постоянно совершенствуются ее как аппаратные, так и программные средства. Функциональность постоянно расширяется, что позволяет поддерживать многие самые новые и передовые технологии.

Реализация необходимых условий для функционирования кластера при использовании сети *InfiniBand*

Обеспечение обмена данными между приложениями в соответствии со стандартом *MPI*. В современных высокопроизводительных кластерных вычислительных системах обмен данными между приложениями, выполняющимися на разных вычислительных узлах, посредством протокола *MPI* уже давно не является функцией *Ethernet*. Как раз для этого и используются высокоскоростные сети, одной из которых и является *InfiniBand*. Следовательно, имеется большая практика успешного ее применения для решения вышеупомянутой задачи. Потому никаких проблем с реализацией данной функции без *Ethernet* возникнуть не должно.

Передача *IP* пакетов. Правила передачи *IP* пакетов через сеть *IB* описывает группа протоколов *IP over InfiniBand (IPoIB)* [3], в соответствии с которыми все сетевые приложения, использующие протоколы TCP/IP в сети *Ethernet*, могут без изменений использоваться в сети *InfiniBand*. Отдельно стоят приложения, работающие с физическими адресами сетевых плат и использующие *Ethernet* пакеты напрямую (примером таких приложений являются *DHCP*-сервер и *DHCP*-клиент), в этих случаях не представляется возможным динамически получать настройки сети с *DHCP*-сервера и создаются некоторые неудобства при сетевой загрузке операционной системы.

Сетевая загрузка операционной системы. Не так давно в рамках проекта *Etherboot/gPXE* [4] была разработана технология, позволившая осуществлять удаленную загрузку операционной системы *Linux* в сетях *InfiniBand*. Она базируется на тех же принципах, что и загрузка через *Ethernet*, но из-за определенных различий в аппаратных средствах имеет свои особенности. Для понимания причины их возникновения рассмотрим сначала классическую *PXE*-загрузку операционной системы *Linux*.

Для загрузки применяются протоколы *IP*, *UDP*, *DHCP* и *TFTP*, осуществляется она загрузчиком *pxelinux*, который можно создать на базе пакета *syslinux* [4]. Кроме того, *BIOS* сетевой платы должен иметь специальный *PXE-код*. *PXE* загрузка операционной системы *Linux* имеет следующие этапы:

1. Посылка запросов *DHCP* серверу *PXE-кодом* сетевой платы и получение начальных настроек сети, а также адреса *TFTP* сервера и пути к загрузчику *pxelinux*.
2. Загрузка с *TFTP* сервера образа *pxelinux* и передача ему управления.
3. Получение с *TFTP* сервера образов ядра *Linux* и временной корневой файловой системы *initrd*, загрузка ядра и монтирование файловой система *initrd*, с последующей передачей управления скрипту инициализации базовой операционной системы.
4. Получение сетевых настроек от *DHCP* сервера и настройка сети.
5. Монтирование по *NFS* основной корневой файловой системы и запуск скрипта инициализации основной операционной системы.

Из этого перечисления следует, что для поддержки сетевой загрузки на оборудовании *InfiniBand* необходимо выполнение следующих условий:

- Плата InfiniBand должна поддерживать технологию PXE.
- TFTP сервер должен работать по протоколу IPoIB.
- NFS также должен работать в соответствии с протоколом IPoIB.
- Должно быть обеспечено динамическое получение настроек IPoIB сетевого интерфейса.
- В ядро Linux должны быть включены драйвера сетевой платы IB, а также поддержка протокола IPoIB.
- Скрипт инициализации базовой операционной системы, входящий в состав initrd, должен быть адаптирован к новому сетевому окружению.

С настройкой *TFTP* и *NFS* никаких проблем не возникает, так как они без всяких изменений работают через *InfiniBand*, используя протокол *IPoIB*. В новых версиях ядер *Linux* присутствуют драйверы для плат *InfiniBand* с поддержкой протокола *IPoIB*, т.е. необходимо собрать ядро *Linux* с этими драйверами. С остальными пунктами все не так просто. Стандартный *BIOS* плат *InfiniBand* не содержит в себе *PXE-кода*. Однако есть возможность это исправить. Для этого необходим образ оригинального *BIOS*, средства для его модификации и перезаписи, а также сам *PXE-код*, который входит в состав программного пакета *Boot over IB* [3]. Все эти компоненты можно получить на сайте производителя сетевого оборудования *InfiniBand*.

Отдельно нужно остановиться на динамическом получении настроек сетевого интерфейса *IPoIB* посредством протокола *DHCP*. Ни стандартный сервер *DHCP*, ни стандартный клиент не поддерживают работу с *InfiniBand*. С помощью патча, входящего в состав пакета *Boot over IB*, можно получить базовую поддержку *InfiniBand* сервером *DHCP*, однако полной совместимости пока нет. К тому же, изменения не касаются клиентской части, а значит, в дальнейшем для полной совместимости необходима доработка как сервера, так и клиента. Связано это с тем, что размер физического адреса плат *InfiniBand* не соответствует протоколу *DHCP*, вследствие чего его нельзя использовать для идентификации *DHCP* сессии. В документации пакета *Boot over IB* предлагается в качестве решения этой проблемы использовать специальный идентификатор, который должен быть записан в конфигурационном файле *DHCP*-клиента. Подобное решение проблемы вполне применимо при загрузке отдельных серверов, но не для загрузки вычислительных узлов кластера, так как все конфигурационные файлы у них являются общими.

Полная совместимость *InfiniBand* и *DHCP* до сих пор является только частично решенной задачей, следовательно, приходится находить локальные решения, как, например, динамическое формирование идентификатора, зависящее от физического адреса сетевой платы, или вообще получать нужные настройки иным путем.

Удаленное управление аппаратными средствами кластера. Наиболее распространенным средством для решения этой задачи является, безусловно, *IPMI* [5]. Все существующие реализации *IPMI* требуют наличия *Ethernet*. Подобных по функциональности средств, способных работать в сети *InfiniBand* не создано и не планируется. Однако возможен альтернативный вариант решения данной задачи – использование сервисной сети *ServNET* [6].

Основная функциональность сервисной сети ServNET:

- селективный сброс узла;
- селективное и «плавное» включение/выключение электропитания узла (предупреждает износ оборудования и позволяет избежать сильного скачка напряжения при включении системы);
- доступ к сериальной консоли узла, поддерживающий: изменение параметров *BIOS* узла; выбор (*LILO*) загружаемой ОС; параметры загрузки ядра *Linux*; любые команды в консольном режиме; мониторинг критических сообщений ОС; «посмертное» чтение (из энергонезависимой памяти платы *ServNET*) нескольких последних сообщений ОС.

Эта сервисная сеть обладает меньшей, чем в *IPMI*, но вполне достаточной функциональностью, она проще в установке, настройке и использовании, кроме того, в силу своей простоты является достаточно стабильной в работе и отказоустойчивой. При всем этом ее стоимость на порядок ниже стоимости *IPMI*. Учитывая все вышеперечисленное, *ServNET* вполне может быть использована для решения задачи удаленного управления аппаратными средствами кластера.

Преимущества и недостатки кластерной вычислительной системы без сети Ethernet

При проектировании высокопроизводительных кластерных вычислительных систем необходимо учитывать, что с ростом производительности системы значительно увеличивается количество необходимого вспомогательного оборудования, в том числе и коммутационного. В действительно больших системах это становится дополнительной проблемой, так как может серьезно повлиять как на стоимость системы, так и на ее отказоустойчивость. Поэтому перспектива уменьшить количество сетевого оборудования и коммутационных кабелей почти в два раза выглядит весьма неплохо. Итак, а в чем же именно мы выигрываем в данной ситуации:

- Увеличивается отказоустойчивость.
- Снижается стоимость.
- Упрощается инсталляция и обслуживание.

К недостаткам можно, пожалуй, отнести только сложность внедрения. Этот недостаток можно объяснить «юным возрастом» технологии, вследствие чего некоторые решения не полностью реализованы. Наиболее сложной из таких проблем является недостаточная совместимость *InfiniBand* и *DHCP*, которая доставляет немало трудностей. Но, тем не менее, даже это не является критичным, к тому же данная проблема может быть решена в процессе дальнейшего развитии технологии.

Выводы

Приведенная концепция построения суперкомпьютеров кластерной архитектуры без использования сети *Ethernet* имеет ряд существенных преимуществ по сравнению с классическими. Учитывая то, что при этом недостатки минимальны, перспективы ее развития выглядят весьма неплохими. Безусловно, имеются некоторые трудности с первоначальным внедрением технологии, но они вполне преодолимы. Практическим подтверждением этого является тот факт, что в 2008 году в Институте кибернетики НАН Украины был модернизирован один из кластеров суперкомпьютерного комплекса *СКИТ* [7], с применением вышеописанных подходов, что позволило полностью отключить его от сети *Ethernet*. В дальнейшем планируется развитие и внедрение данной технологии в кластерных решениях Института кибернетики НАН Украины.

Литература

1. Режим доступа: http://parallel.ru/tech/tech_dev/mpi.html
2. Режим доступа: http://parallel.ru/computers/interconnects.html
3. Режим доступа: http://www.mellanox.com
4. Режим доступа: http://www.etherboot.org
5. Режим доступа: http://www.intel.com/design/servers/ipmi/ipmi.htm
6. Режим доступа: http://www.t-platforms.ru
7. Режим доступа: https://icybcluster.org.ua/

С.О. Горенко, А.Л. Головинський, С.Г. Рябчун

Побудова суперкомп'ютера кластерної архітектури без використання мережі Ethernet

Досліджені загальні концепції управління високопродуктивною кластерною обчислювальною системою без використання мережі Ethernet, а також розглянута можливість практичної реалізації такого управління.

S.A. Gorenko, A.L. Golovinsky, S.G. Ryabchun

Supercomputer Building with the Cluster Architecture without Usage of Ethernet

The general concepts of construction high-efficiency cluster computing system without usage of network Ethernet are researched, and also the opportunity of their practical realization is considered.

Статья поступила в редакцию 22.07.2008.